eBMU

BlueMoon™ Universal Embedded

SPP-AT Application

eBMU PMB 8753/2 Version 1.01
eBMU SPP-AT Image

User's Manual

Software Description
Revision 5.0, 2011-09-05

Intel Mobile Communications

Intel Corporation has recently acquired certain portions of Infineon Technologies AG and is in the process of converting documents to the Intel format. Please bear with us during this transition and direct any questions or issues to your sales rep or to IMC-DOC.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined”. Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: **http://www.intel.com/#/en_US_01**.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

[When the doc contains software source code, include a copy of the software license or a hyperlink to its permanent location.]

This document contains information on products in the design phase of development.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: **http://www.intel.com/products/processor%5Fnumber/**.

Code Names are only for use by Intel to identify products, platforms, programs, services, etc. (“products”) in development by Intel that have not been made commercially available to the public, i.e., announced, launched or shipped. They are never to be used as “commercial” names for products. Also, they are not intended to function as trademarks.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Core Inside, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, the Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, InTru, the InTru logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, skoool, the skoool logo, Sound Mark, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2011, Intel Corporation. All rights reserved.

eBMU

Revision History: 2011-09-05, Revision 5.0

Previous Revision: 2010-04-28, Rev. 4.1

Page	Subjects (major changes since last revision)
16	Length of uuid parameter in AT+JRLS command depends on the used SPP-AT SW version.
20	The timing of Stream Connection Cancel string is more tolerant in SPP-AT v1.7 and onwards.
25-26	New section 4.11 GPIO Indication of Connection Status

Trademarks of Infineon Technologies AG

BlueMoon™, COMNEON™, C166™, CROSSAVE™, CanPAK™, CIPOS™, CoolMOS™, CoolSET™, CORECONTROL™, DAVE™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPACK™, EconoPIM™, EiceDRIVER™, EUPEC™, FCOS™, HITFET™, HybridPACK™, ISOFACE™, I²RF™, IsoPACK™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OmniTune™, OptiMOS™, ORIGA™, PROFET™, PRO-SIL™, PRIMARION™, PrimePACK™, RASIC™, ReverSave™, SatRIC™, SensoNor™, SIEGET™, SINDRION™, SMARTi™, SmartLEWIS™, TEMPFET™, thinQ!™, TriCore™, TRENCHSTOP™, X-GOLD™, XMM™, X-PMU™, XPOSYS™.

Other Trademarks

Advance Design System™ (ADS) of Agilent Technologies, AMBA™, ARM™, MULTI-ICE™, PRIMECELL™, REALVIEW™, THUMB™ of ARM Limited, UK. AUTOSAR™ is licensed by AUTOSAR development partnership. Bluetooth™ of Bluetooth SIG Inc. CAT-iq™ of DECT Forum. COLOSSUS™, FirstGPS™ of Trimble Navigation Ltd. EMV™ of EMVCo, LLC (Visa Holdings Inc.). EPCOS™ of Epcos AG. FLEXGO™ of Microsoft Corporation. FlexRay™ is licensed by FlexRay Consortium. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. MATLAB™ of MathWorks, Inc. MAXIM™ of Maxim Integrated Products, Inc. MICROTEC™, NUCLEUS™ of Mentor Graphics Corporation. Mifare™ of NXP. MIPI™ of MIPI Alliance, Inc. MIPS™ of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO., MICROWAVE OFFICE™ (MWO) of Applied Wave Research Inc., OmniVision™ of OmniVision Technologies, Inc. Openwave™ Openwave Systems Inc. RED HAT™ Red Hat, Inc. RFMD™ RF Micro Devices, Inc. SIRIUS™ of Sirius Sattelite Radio Inc. SOLARIS™ of Sun Microsystems, Inc. SPANSION™ of Spansion LLC Ltd. Symbian™ of Symbian Software Limited. TAIYO YUDEN™ of Taiyo Yuden Co. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. TOKO™ of TOKO KABUSHIKI KAISHA TA. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VLYNQ™ of Texas Instruments Incorporated. VXWORKS™, WIND RIVER™ of WIND RIVER SYSTEMS, INC. ZETEX™ of Diodes Zetex Limited.

Last Trademarks Update 2010-03-22

Table of Contents

1 Introduction 5

2 Bluetooth Features 6

3 Serial Port Profile 7
3.1 Operation Modes 7
3.1.1 Command Mode 7
3.1.2 Stream Mode 7
3.2 Pin Assignments 7

4 Infineon SPP-AT Command and Response 9
4.1 AT Command and Response Format and Syntax 9
4.1.1 AT Commands 9
4.1.2 AT Responses 9
4.1.2.1 AT Response Parameter List for <status> and <ERROR> 10
4.1.2.1.1 Command Execution Status Values 10
4.1.2.1.2 General Error Messages 10
4.2 AT Command List Table 11
4.3 AT Responses List (not command triggered) 21
4.4 Crystal Auto Calibration (Frequency counter method) 22
4.5 Low Power Mode Control 22
4.5.1 Host Initiates Low Power Mode Entry and Exit 23
4.5.2 Host Initiates Low Power Mode Entry, eBMU Initiates Exit 23
4.6 UART Baud Rate Change 24
4.7 Data Flow Control 25
4.8 Production Mode 25
4.9 SPP-AT Commands Operating with EEPROM 25
4.10 Security Mode 25
4.11 GPIO Indication of Connection Status 25

5 Example AT Commands and Responses 27

References 30

Terminology 31

1 Introduction

This document describes the AT commands applicable to Infineon's eBMU Bluetooth solution for embedded applications. The product utilizes a raw AT set of commands and responses over UART for Bluetooth communications and control on an embedded device.

2 Bluetooth Features

- Bluetooth v2.0 + EDR compliant.
- Device A and B support 1 ACL link with stream or command mode.
- Device A and B - Visible while connected.
- Device A and B - Visible/connectable when not connected.
- Device A and B - Device Discovery capable after receiving OK on data transfer.
- Sniff mode is supported with above capabilities.
- 5 trusted devices stored in EEPROM.
- Testing
- Enable DUT.
- Crystal calibration.
- H4 with UART HW flow control (RTS/CTS).
- Security modes
- Mode 1 and Mode 3.
- Master-Slave role switch

3 Serial Port Profile

Host communication sent over UART is always called command except while in stream mode, see below. All communication received by host application over UART is called response except while in stream mode.

3.1 Operation Modes

The specification defines two operation modes of the PMB 8753/2 also named eBMU throughout the document. For Bluetooth SPP there are two different roles specified; A-device (paging device) setting up the over the air connection and B-device which is connectable and accepts the connection (page scanning device).

3.1.1 Command Mode

In this mode the SPP application running on the eBMU will execute the AT commands sent from the host over the UART using the H4 UART protocol as specified within the Bluetooth SIG **[1]**. In this mode, the host application can send data to the eBMU, which are transmitted to the remote device which has a Bluetooth connection on SPP level with the eBMU. This mode is normally used when transmitting burst and packetized data.

Setting up/accepting Bluetooth SPP connections and/or searching for other Bluetooth devices are also operations done in this mode.

3.1.2 Stream Mode

In this mode, the host application will send un-packetized data to the eBMU, which are transmitted over the air to the remote device. This mode is normally used when transmitting small size of data in a random way and for serial cable replacement applications.

3.2 Pin Assignments

The table below shows the available GPIOs. GPIOs noted with “(Reserved)” can not be controlled by a host. P0.1 and P0.8 can be used as application GPIOs but not when they are used during the crystal calibration procedure. All application GPIOs are by default configured as tri-state.

Pin Name	Default Configuration	Direction	GPIO	Description
UART_RX	I	I	P0.5 (Reserved)	UART
UART_TX	O PU	O	P0.4 (Reserved)	UART
UART_RTS	O PU	O	P0.6 (Reserved)	UART
UART_CTS	I	I	P0.7 (Reserved)	UART
SDA	O PU	I/O	P0.12 (Reserved)	I2C
SCL	I PU	I	P0.13 (Reserved)	I2C
PCMFR1	O PU	O	P0.0 (Reserved)	LPM
TX_Conf2	I	I	P0.14 (Reserved)	LPM
PCMCLK	Z	I/O	P0.1	APPL GPIO / XTAL CAL
PCMIN	Z	I/O	P0.2	APPL GPIO
PCMOUT	Z	I/O	P0.3	APPL GPIO
PAON	Z	I/O	P0.8	APPL GPIO / XTAL CAL
PSEL0	Z	I/O	P0.9	APPL GPIO
PSEL1	Z	I/O	P0.10	APPL GPIO
TX_Conf1	Z	I/O	P0.11	APPL GPIO
P015	Z	I/O	P0.15	APPL GPIO

4 Infineon SPP-AT Command and Response

AT commands can only be sent while in command mode with the exception of stream connection cancel (^^^) which can be sent in stream mode. The expected response after sending an AT command is the “OK” response, see specification below. The host shall wait for a command to be terminated before sending a new one. A command is considered as terminated when “OK” and all subsequent related responses have been received.

There are also responses, which are not initiated by a sent command. They are in that case initiated by the remote Bluetooth device.

4.1 AT Command and Response Format and Syntax

All data exchanged between the host and eBMU is in ASCII format.

Parameters for commands and responses are given in decimal (DEC) base in ASCII format unless hexadecimal (HEX) base is specified.

MSB is always sent first. eBMU does not distinguish between upper and lower cases.

Example for values with decimal base in ASCII format

The number 255 in decimal corresponds to three characters ‘2’, ‘5’, ‘5’.

E.g. AT+JSDA=010,1234567890. Here the number 10 must be given by three ASCII characters ‘0’, ‘1’, ‘0’, because the parameter requires a value consisting of three characters.

Example for values with hexadecimal base in ASCII format

The number 255 in decimal base corresponds to the number FF in hexadecimal base, thereby the number in ASCII format for hexadecimal base is represented by the following two characters ‘F’, ‘F’.

E.g. +RSNFCNF=3E80,2. The four ASCII characters ‘3’, ‘E’, ‘8’, ‘0’ represent the number 3E80 in hexadecimal base which corresponds to 16000 in decimal base.

4.1.1 AT Commands

All AT-commands follow the format below;

AT+<command>=<parameter 1 (if required)>,<parameter 2 (if required)>,<parameter 3 (if required)>,<…><carriage return><line feed>

E.g.: AT+JCCR=0010c64d67dc,01 (To connect to BD_ADDR 0x0010c64d67dc, service channel 1)

4.1.2 AT Responses

All AT-responses follow the format below with the exception of <OK> and <ERROR=>;

+<response>=<parameter 1 (if required)>,<parameter 2 (if required)>,<parameter 3 (if required)>,<…><carriage return><line feed>

E.g.: +RDAI =004,DATA (4 bytes “DATA” received)

4.1.2.1 AT Response Parameter List for <status> and <ERROR>

4.1.2.1.1 Command Execution Status Values

Values for <status> general for all commands

- BT_OK 0
- BT_ERROR 1
- BT_TIMEOUT 4

4.1.2.1.2 General Error Messages

- ERR = -1; Syntax Error
- ERR = -2; Command not allowed at present execution status
  - ERR=-2 is obtained when any of the following situations is met:

1. Create a new connection when already connected.
2. Device discovery when already connected.
3. Send data when not connected.
4. Accept connection request without a request.
5. Enter sniff mode without being connected.
6. Enable SEC mode after another command has previously been sent.
7. Service discovery when connected.
8. Send data with length = 0.

4.2 AT Command List Table

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
AT+JAAC=<*auto_accept*>	***Auto Accept Connection requests*** Forces eBMU to accept connection requests. ***auto_accept*** parameter (1 character): 0 Host will be notified on incoming connection indication (*+RCOI*) – (no auto accept). Default value 1 eBMU will automatically auto accept incoming connection request – (host will be notified but connection is accepted automatically)	**OK**	Dev B
AT+JACR=<*accept*>	***Accept Connection Request*** Shall be used as answer to a connect indication (*+RCOI*). ***accept*** parameter (1 character): 0 Not accepted 1 Accepted	**OK**	Dev B
AT+JCAC=<*trim_value*>, <*GPIO* >	***Crystal Auto Calibrate*** ***trim_value*** parameter (4 characters / HEX base): The trim value is used to adjust the frequency on the GPIO chosen by the GPIO parameter. ***GPIO*** parameter (4 characters / HEX base): Two GPIOs can be used as output for the oscillator trim, P01 and P08 (0002 and 0100). See section **4.4**. *Note: **AT+JCAC** command can only be issued after production mode is enabled (**AT+JPRO=1**). See section **4.8**.*	**+RCACCNF=<*osc_trim*>** ***osc_trim*** parameter(4 characters / HEX base) Use **AT+JCBD** to store the *osc_trim* value to EEPROM.	Dev A Dev B
AT+JCBD=<*bd_data*>	***Change BD_Data*** ***bd_data*** parameter (116 characters / HEX base / LSB first): The *bd_data* string consists of following configuration data (with reference value): bdAddr (12 characters) = see Note 1 channelWordOffset (4 characters) = 0000 clkConf (2 characters) = 8A	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	eepromSize (2 characters) = 80 inputFreq (8 characters) = 80BA8C01 lmpFeatures (16 characters) = BF02067E98190000 lpmConf (2 characters) = 40 lpmDrift (2 characters) = FA lpmThreshold (2 characters) = 12 ulpmThreshold (2 characters) = 18 pmuConfig (4 characters) = 8000 rfPselD (8 characters) = 03040506 rfPselConf (2 characters) = 44 rssiMin (2 characters) = 0C rssiMax (2 characters) = 10 ddcTlConf (2 characters) = 02 uartBaudrate (2 characters) = 04 uartInvert (2 characters) = 00 uartPulls (2 characters) = 01 oscSettle (2 characters) = 10 bbConf (2 characters) = 04 rfConf (2 characters) = 04 txPowerRef0 (2 characters) = F2 txPowerRef1 (2 characters) = F8 txPowerRef2 (2 characters) = FE txPowerRef3 (2 characters) = 04 oscTrim (4 characters) = see Note 2 threeWireArqTimeout (2 characters) = 06 lmpVersion (2 characters) = 00 reserved (16 characters) = 0000000000000000 *Note:* 1. *The bdAddr is pre-programmed on eUniStone module. It can be read through **AT+JRBD**. Don't overwrite the pre-programmed value accidentally.* 2. *The oscTrim value is pre-programmed on eUniStone module. It can be read through **AT+JRBD**. Writing a wrong value can destroy the module.* 3. ***AT+JCBD** command can only be issued after production mode is enabled (**AT+JPRO=1**). Changes in BD Data take place after production mode command is disabled (**AT+JPRO=0**) followed by a SW reset (**AT+JRES**). See section **4.8**.*		

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
AT+JCCR=<*bd_addr*>, <*service channel*>	***Create Connection Request*** Instructs eBMU to connect to a remote Bluetooth device (prospective slave). ***bd_addr*** parameter (12 characters / HEX base): The Bluetooth address of the remote device ***service_channel*** parameter (2 characters / DEC base): Value range: 01-30 Which service channel to connect to can be received from a Service Discovery (**AT+JSDS**)	**OK** (Followed by): **+RCCRCNF=**<***MTU_size***>,<***status***> ***MTU_size*** parameter (3 characters / DEC base) ***status*** parameter (1 character): See section 4.1.2.1.1. If maximum number of allowed connections already exists: **ERR=-2**.	Dev A
AT+JDDS=<*limit_inquiry_result*>	***Device Discovery Start*** Causes eBMU to start a Device Discovery (Inquiry and Remote Name Request) of the Bluetooth neighborhood. ***limit_inquiry_result*** parameter (1 character): Value range: 0-1. The maximum number of responses that can be obtained is 8. *Note: Value of the limit_inquiry_result parameter have no impact in the behavior of the command.*	**OK** (Then, if responses are returned): **+RDDSRES=**< ***bd_addr*** >, <***remote_name***>,<***COD***> (For each response) ***bd_addr*** parameter (12 characters / HEX base). ***remote_name*** parameter (variable length): Name of the remote device. Page Timeout is received as a name if the remote name is not found during device discovery ***COD*** parameter (6 characters / HEX base): Class of device (Completed by): **+RDDSCNF=**<***status***> ***status*** parameter (1 character): See section 4.1.2.1.1.	Dev A
AT+JDIS=<*discoverable*>	***DIScoverable*** Forces eBMU into Page Scan / Inquiry Scan states indefinitely (note: this makes the device discoverable). ***discoverable*** parameter (1 character): 0 No scans enabled. 1 Inquiry Scan enabled.	**OK**	Dev B Dev A is discover-able but does not accept incoming con-nections.

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	2 Page Scan enabled. 3 Inquiry & Page Scan enabled. *Note: Default value after a HW reset is no scans enabled*		
AT+JDOI	***DOwnload Application Image via UART*** After "OK" response a binary file with the EEPROM image should be sent. File open and send in binary format after "*OK*" is received. *Note: **AT+JDOI** command can only be issued after production mode is enabled (**AT+JPRO=1**). Changes in the EEPROM image are effective after production mode command is disabled (**AT+JPRO=0**) followed by a HW reset. See section **4.8.***	**OK** after command is sent. **+RDOICNF** after .eep file has been written. *Note: eBMU calculates the size of the image that is being downloaded from the file header. The response +RDOICNF is generated when the calculated size is reached.*	Dev A Dev B
AT+JEDT	***Enable Device under Test*** This SPP-AT command enables the device under test. After this command has been sent it is possible for a remote tester to connect, this AT command corresponds to the three different HCI commands listed below: 1. Set Event Filter – allow all connections. 2. Write Scan Enable – page and Inquiry. 3. Enable device under test. *Note: **AT+JEDT** command can only be issued with security mode 1 and after production mode is enabled (**AT+JPRO=1**). See section **4.8**.*	**OK**	Dev A Dev B
AT+JGPA=<*reserved*>, <*read*>, <*set*>,<*clear*>	***GPIO action*** All parameters of this command are bit fields of 16 bits corresponding to GPIOs P0.15 to P0.0. E.g. if pin P0.0 is the desired bit; the bit field value is 0001 and if the desired bit is P0.12 the bit field value is 0800. ***reserved*** parameter (4 characters) ***read*** parameter (4 characters / Hex base): Values for each bit: 0 No Action 1 Read	**+GPOACNF=<*value*>** ***value*** parameter (4 characters / HEX base): This value is the state of the GPIO PINs specified in read parameter. Values for each bit: 0 Low 1 High When the pins are set as outputs the return	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	set parameter (4 characters / HEX base): Values for each bit: 0 No Action 1 Set ***clear*** parameter (4 characters / HEX base): Values for each bit: 0 No Action 1 Clear	value will be 0 for the specific pin. Reserved pins will always return 0.	
AT+JGPC= ***<direction>,<open_drain>, <pull_on/off>, <pull_up/down>,<tristate>***	***GPIO Configuration*** All parameters of this command are bit fields of 16 bits corresponding to GPIOs P0.15 to P0.0 (See command **AT+JGPA**). ***direction*** parameter (4 characters / HEX base): Values for each bit: 0 OUT 1 IN ***open_drain*** parameter (4 characters / HEX base): Values for each bit: 1 OPEN ***pull_on/off*** parameter (4 characters / HEX base): Values for each bit: 0 Pull OFF 1 Pull ON ***pull_up/down*** parameter (4 characters / HEX base): Value for each bit: 1 Pull UP 0 Pull DOWN ***tristate*** parameter (4 characters / HEX base): Value for each bit: 1 Tri-state *Note: See section **3.2** for available GPIO pins.*	**OK**	Dev A Dev B
AT+JPCR=*<length_PIN_code>, <PIN_code>*	***PIN Code Reply*** Sent to eBMU in response to a PIN Code Request from a remote Bluetooth device (bd_addr). ***length_PIN_code*** parameter (2 characters / DEC base): Value range: 01-16 Length of PIN code	**OK** (Followed by): **+RSLE** if secure link is established	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	PIN_code parameter (length= *length_PIN_code*): The PIN Code to be sent to the remote Bluetooth device, e.g. **AT+JPCR**=04,1234 *Note: The PIN code is an ACII string.*		
AT+JPRO=<*mode*>	***PROduction mode*** ***mode*** parameter (1 character): 0 Production mode OFF 1 Production mode ON *Note: See section **4.8** for commands requiring production mode.*	**OK**	Dev A Dev B
AT+JRBD	***Read Bluetooth device Data*** Sent to eBMU to retrieve the Bluetooth Device Address and the Oscillator trimming value.	**+RRBDRES=<*bd_addr*>, < *oscTrim*>** ***bd_addr*** parameter (12 characters / HEX base): BD Address of remote device ***oscTrim*** parameter (4 characters / HEX base) Oscillator trim value	Dev A Dev B
AT+JRES	***RESet*** SW reset of the system.	**ROK** as applications is restarted.	Dev A Dev B
AT+JRLS=<*uuid*>, <*length_ service_name* > <*service_name*>,<*service channel* >, <*CoD*>	***Register Local Service*** ***uuid*** parameter: uuid for supported profile With SPP-AT v1.x: 4 characters / HEX base e.g.1101 for Serial Port Profile With SPP-AT v2.x: 32 characters / HEX base e.g. 00001101000010008000000805f9b34fb for Serial Port Profile ***length_service_name*** parameter (2 characters / DEC base): Value range: 01-16 Length of service name ***service_name*** parameter (length= length_service_name): Name for the service, no final delimiter is needed ***service channel*** parameter (2 characters / DEC base): Value range: 01-30 Which service channel to connect to can be	**OK**	Dev B

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	received from a Service Discovery with **AT+JSDS**. ***CoD*** parameter (6 characters / HEX base): Class of device. The default CoD value is 000000 *Note:* *1. You may not register more than one service.* *2. This command should be sent if DevB behavior is required.*		
AT+JRRI	***Read Revision Information***	<***revision***> ***revision*** parameter (2 characters / HEX base)	Dev A Dev B
AT+JRTD=<*bd_addr*>	***Remove Trusted Device*** Delete the trusted device information for a registered device (*bd_addr*). ***bd_addr*** parameter (12 characters / HEX base): The Bluetooth address of the device that shall be removed from the list. <*bd_addr*>=000000000000: Delete all trusted devices.	**OK**	Dev A Dev B
AT+JSCR	***Stream Connection Request*** Connects the SPP and UART streams, transparent communication will be enabled if both sides execute this command.	**OK**	Dev A Dev B
AT+JSDA=<***length***>, <***data***>	***Send Data Request*** ***length*** parameter (3 characters / DEC base): number of bytes to be sent Value range: 001 to max .MTU size ***data*** parameter (see note for size): Data to be sent *Note: Maximum number of bytes for each packet is reported at connection confirmation (MTU_Size). The value for this parameter is negotiated by the two devices during connection set up.*	**OK**	Dev A Dev B
AT+JSDR	***SPP Disconnect Request*** Forces an SPP disconnection.	**OK**	Dev A Dev B
AT+JSDS=<***bd_addr***>, < ***uuid***>	***Service Discovery Start*** Causes eBMU to start a service discovery of device with bd_addr and search for services defined by uuid.	**OK** (Then, if services are returned): **+RSDSRES=**	Dev A

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	bd_addr parameter (12 characters / HEX base): BD Address of remote device ***uuid*** parameter (4 characters / HEX base): Service to search for e.g. 1101 for Serial Port Profile	<***remote_service_name***>, <***remote_service_channel***> ***remote_service_name*** parameter (variable length): Name of the remote service. ***remote_service_channel*** parameter (2 characters / DEC base) (For each service) (Completed by): **+RSDSCNF=**<***status***> ***status*** parameter (1 character)	
AT+JSEC= ***<security_mode>,*** ***<Link_key_information>,*** ***<PIN_type>,*** ***<length_PIN_code>,*** ***<PIN_code>***	***Enable SECurity*** ***security_mode*** parameter (1 character): 1 Security Mode 1 2 N/A 3 Security Mode 3 (default) ***link_key_information*** parameter (1 character): 1 - 2 - *Note: This parameter has no function. It has to be set to either 1 or 2.* ***PIN_type*** parameter (1 character) 1 Variable PIN (default) 2 Fixed PIN ***length_PIN_code*** parameter (2 characters / DEC base): Length of PIN code. The maximum PIN length value is 16 (corresponding to a 16 characters long PIN code) ***PIN_code*** parameter (length=length_PIN_code): Normal user PIN, for example “0000” (default), This parameter is taken into account if PIN_TYPE is fixed. *Note: If security mode shall be changed from default settings, **AT+JSEC** shall be the first command that is sent after a reset. Security mode 3 is the default security mode and the default PIN*	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	type is variable.		
AT+JSLN= <***length_friendly_name***>, <***friendly_name***>	***Set Local device friendly Name*** Supports all ASCII characters. ***length_friendly_name*** parameter (2 characters / DEC base): Length of friendly name, the maximum value for length is 18. ***friendly_name*** parameter (length=*length_friendly_name*): No delimiter is required.	**OK**	Dev A Dev B
AT+JSNF=<***sniff_Max***>, <***sniff_Min***>, <***sniff_attempt***>, <***sniff_tmo***>,<***on/off***>	***Sniff Request*** Request a link to enter Sniff Mode. All command parameters are given in HEX base with the exception of on/off parameter. 1 ASCII character represents four bits in HEX base e.g. “A” is 1010. MSB given first. ***sniff_Max*** parameter (4 characters / HEX base): Maximum allowed sniff interval Value to be written= N Time = N * 0.625 ms Range: 1.25 ms to 40.9 s ***sniff_Min*** parameter (4 characters / HEX base): Minimum allowed sniff interval Value to be written= N Time = N * 0.625 ms Range: 1.25 ms to 40.9 s ***sniff_attempt*** parameter (4 characters / HEX base): Number of sniff attempts Value to be written= N Length = N* 1.25 ms Time Range: 0.625 ms - 40.9 s ***sniff_tmo*** parameter (4 characters / HEX base): The time out value for sniff attempts Value to be written= N Time = N * 0.625 ms Range: 0 msec to 40.9 s ***on//off*** parameter (1 character): Sniff ON (value=1) or Sniff OFF (value=0)	**+RSNFCNF=** <***sniff_Interval***>,<***mode***> ***sniff_Interval*** parameter (4 characters / HEX base): ***mode*** parameter (1 character): Normal mode (mode=0) or sniff mode (mode=2) If no connection exists: **ERR=-2**.	Dev A Dev B
^ ^ ^	***Stream Connection Cancel*** Exit Streaming Mode	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	Dev A / Dev B
	Send -T0 - ^ -T1- ^ -T1- ^ -T0- with an interval of 100 ms < T1 < 1100 ms between symbols and T0 > 100 ms.[1] *Note: This string is not terminated with CR LF.*		

[1] If SPP-AT v1.0 to v1.6, or v2.0 is used, send -T0 - ^ -T1- ^ -T1- ^ -T0- with an interval of T1= 1 second between symbols and T0 > 1 second.

4.3 AT Responses List (not command triggered)

AT Response	Usage	Command to acknowledge the response	Dev A / Dev B
ROK	***Start up response***		Dev A Dev B
+RPCI=<***bd_addr***>	***PIN Code Indication*** ***bd_addr*** parameter (12 characters / HEX base): MSB is sent first.	**AT+JPCR=**<***length_PIN_code***>, **<*PIN_code*>** ***length_PIN_code*** parameter (2 characters / DEC base): Length of PIN code ***PIN_code*** parameter (length=length_PIN_code):	Dev A Dev B
+RCOI=<***bd_addr***>	***Connect Indication*** ***bd_addr*** parameter (12 characters / HEX base): See previous response for information on the format.	**AT+JACR=** <***accept***> ***accept*** parameter (1 character): 0 Not accepted 1 Accepted	Dev B
+RDAI=<***length***>,<***data***>	***Data Indication*** ***length*** parameter (3 characters / DEC base): Number of bytes to be sent ***data*** parameter (length=*length* from previous parameter) Received data	Not available during stream mode.	Dev A Dev B
+RDII	***Disconnect Indication***	Received on the side that has not initiated the disconnection. During stream mode, if the link is lost, the response **+RDII** is received after the link supervision time out of 20 seconds.	Dev A Dev B
+RSLE	***Secure Link Established***		Dev A Dev B
+RSNFCNF=<***sniff_Interval***>, <***mode***>	***Sniff mode confirmation*** ***sniff_Interval*** parameter (4 characters / HEX base). ***mode*** parameter (1 character): Normal mode (mode=0) or sniff mode (mode=2)	Received on the side that has not issued the sniff command (**AT+JSNF**)	Dev A Dev B
+RRSW	***Role Switch Indication***	Received when the remote device performs a master/slave role switch.	Dev A Dev B

4.4 Crystal Auto Calibration (Frequency counter method)

The frequency of the eBMU's crystal oscillator can be calibrated with a built in capacitance array that is configured by the BD_DATA value Osc_trim.

Instrument required is a frequency counter, for example Agilent 53131A universal counter. This method also requires that one of the GPIO pins P01 or P08 is available.

The calibration will be done on a 32 MHz clock generated from the 26 MHz internal reference clock. This 32 MHz clock should be adjusted to be within ±2ppm (±64 Hz) accuracy.

The command requires that the device is in production mode, please follow the sequence below:

1. Connect the frequency counter to the appropriate test point
2. Enter Production Mode AT+JPRO=1
3. Use The AT+JCAC command to define the test point and the Osc_trim value
4. Measure the frequency of the 32 MHz signal with the counter
5. Iterate steps 3 and 4 until ±2 ppm are reached. Then write the corresponding trim value to the parameter Osc_trim in the BD-data with AT+JCBD
6. Leave production mode AT+JPRO=0
7. Perform a SW reset

4.5 Low Power Mode Control

The low power mode protocol for eBMU is based on hardware signaling only. No SPP commands or responses are required. The existing flow control signals for the UART are used together with two GPIOs to tell the other device (host or controller) when it may enter low power mode, when it should wake up and when it cannot transmit because the first device is in low power mode.

Low power mode can be used when the device is in:

- Page-/inquiry-scan mode
- Connected with link in sniff
- Disconnected

In all other states low power mode shall not be used, a few examples of other states are listed below:

- During connection set-up
- During device discovery
- ACL link without sniff

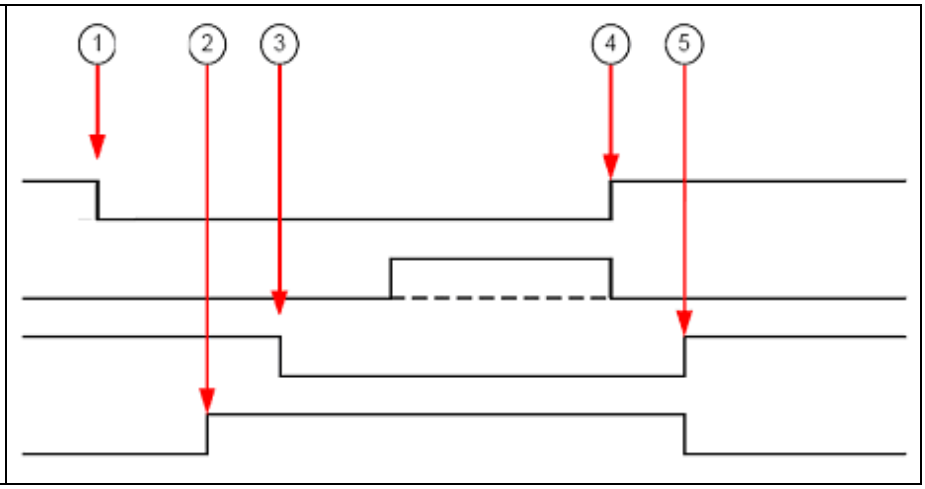
To allow the eBMU to enter low power mode, the host sets PIN P0.14 low. When eBMU is ready, it will also allow the host to enter LPM by setting P0.0 low. Before entering LPM, the host shall set UART CTS of eBMU high. Before entering LPM, eBMU will set its own UART RTS high.

The host can wake up eBMU by setting UART CTS of eBMU low again and setting P0.14 high again. eBMU can wake up the host by setting its own UART RTS low again and setting P0.0 high again.

4.5.1 Host Initiates Low Power Mode Entry and Exit

The picture below describes when the host initiates Low Power Mode and host initiates the Low Power Mode.

Host		eBMU
Host Output	→	GPIO 0.14
Host RTS	→	UARTCTS
Host Input	←	GPIO 0.00
Host CTS	←	UARTRTS

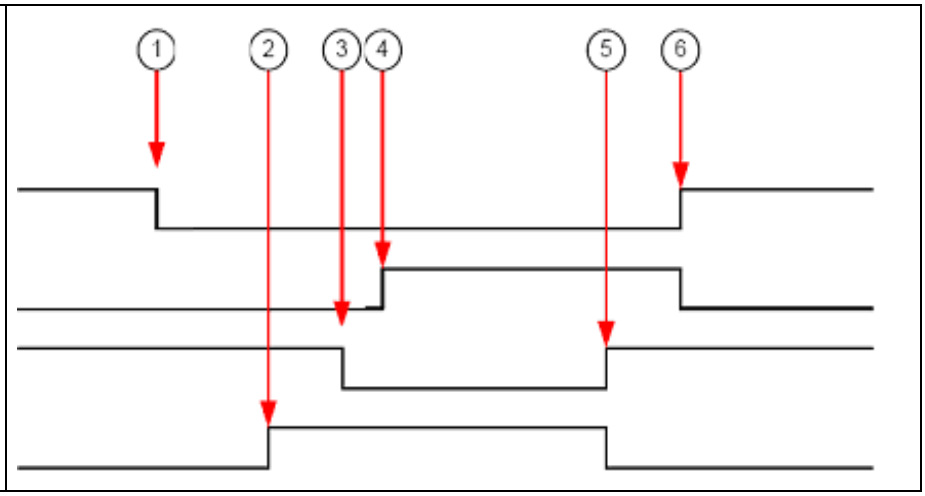


1. The host allows eBMU to enter low power mode
2. eBMU enters low power mode
3. eBMU allows the host to enter low power mode, the host may, if it can, enter low power mode
4. The host requests the eBMU to wake up
5. eBMU wakes up

4.5.2 Host Initiates Low Power Mode Entry, eBMU Initiates Exit

The picture below describes when the host initiates Low Power Mode and host initiates the Low Power Mode.

Host		eBMU
Host Output	→	GPIO 0.14
Host RTS	→	UARTCTS
Host Input	←	GPIO 0.00
Host CTS	←	UARTRTS



1. The host allows eBMU to enter low power mode
2. eBMU enters low power mode
3. eBMU allows the host to enter low power mode
4. The host enters low power mode
5. eBMU requests the host to wake up
6. The host wakes up

4.6 UART Baud Rate Change

The baud rate of the UART can be changed permanently by writing into the UART_Baudrate parameter of the BD_DATA.

The available UART baudrates are:

UART Baud Rate (bauds/s)	Configuration value for BD_DATA
9600	0
19200	1
38400	2
57600	3
115200	4
230400	5
460800	6
921600	7
1843200	8
3250000	9

Procedure to change UART Baud Rate:

1. Read the module specific BD_ADDR and OSC_Trim values and store the values.
   e.g. BD_ADDR=0x000319052BE4, OSC_Trim=0x01D6

   `--> AT+JRBD`

   `<-- +RRBDRES=000319052BE4,01D6`
2. Enter production mode.

   `--> AT+JPRO=1`

   `<-- OK`
3. Change UART baudrate using the “Change BD_Data” command by specifying the corresponding configuration value in BD_DATA. The original BD_ADDR and OSC_Trim values on the module must also be given in the BD_DATA parameter.
   e.g. Change baudrate to 921600

   `-->AT+JCBD=`**`E42B05190300`**`00008A8080BA8C01BF02067E9819000040FA1218800003040506`
   `440C1002`**`07`**`0001100404F2F8FE04`**`D601`**`06000000000000000000`

   `<-- OK`

   Note: The BD_DATA parameter of the AT+JCBD command is composed in the “LSB first” format.
4. Exit production mode.

   `--> AT+JPRO=0`

   `<-- OK`
5. SW Reset

   `--> AT+JRES`

   `<-- ROK` (sent with new baudrate)

The configuration will be written to EEPROM after the “AT+JCBD” command. The EEPROM access will be terminated before the “OK” response. **Power off during that time can corrupt the EEPROM data.**

After this configuration, the module will always use the new baudrate.

4.7 Data Flow Control

eBMU acknowledges a data packet sent by send data command (AT+JSDA) with an “OK” response when it has been transmitted. The host shall wait for the acknowledgement before sending a new packet.

4.8 Production Mode

Production mode is used for configuration and test purposes, the production mode shall be entered in order to execute the following commands:

- AT+JDOI: DOwnload EEPROM Image
- AT+JEDT: Enable Device under Test
- AT+JCAC: Crystal Auto Calibration
- AT+JCBD: Change BD_Data

General procedure to use production mode related commands:

1. HW reset
2. Enter production mode (AT+JPRO=1)
3. Execute command (e.g. “AT+JDOI”)
4. Exit production mode (AT+JPRO=0)
5. HW reset

4.9 SPP-AT Commands Operating with EEPROM

The following commands are related to operations (read/write) in the EEPROM:

1. AT+JDOI (Download Image): It writes the whole content of the EEPROM.
2. AT+JCBD (change BD_DATA): It writes the BD_DATA section of the EEPROM.
3. AT+JRTD (Remove Trusted Device). It erases the associated BD_ADDRESS and link-key from the EEPROM.
4. AT+JRLS (Register Local Service): It writes the CoD value into EEPROM.
5. AT+JCCR (Create connection with security mode 3): The resulting link key will be written into EEPROM (both in DEVA and DEVB).
6. AT+JRES (RESet): The content of EEPROM is read and loaded into RAM.
7. AT+JSLN (Set Local Name): The local name is written into EEPROM.

Note: AT+JRBD does not read from EEPROM, but from the configuration that has been read into the device after startup.

4.10 Security Mode

Pairing is done automatically when security mode 3 is used.

Security mode 3 is the default security mode. If security mode 1 (no authentication and no encryption) is the desired mode, then AT+JSEC should be the first command issued specifying security mode 1 after a HW reset.

The pair procedure involves a PIN code indication +RPCI and a secure link established response +RSLE.

If a trusted device shall be removed, then the remove trusted device command (AT+JRTD) shall be used.

See example in section 5 for a detailed session description.

4.11 GPIO Indication of Connection Status

The GPIO pin P0.1 (pin E5) is used to indicate the connection status. P0.1 is HIGH when BT device is connected and LOW when there is no connection. The transition from HIGH to LOW happens prior to sending

the “+RDII” indication via UART. Hosts that can not monitor the incoming data stream for the "+RDII" indication in stream mode can monitor P0.1 instead.[1]

For backwards compatibility P0.1 is configured as input pin by default. To use this feature the host must send the AT command “*AT+JGPC=FFFD,0000,0000,0000,FFFD*” which configures P0.1 as an output pin.

[1] This feature is available in SPP-AT v1.6 and onwards.

5 Example AT Commands and Responses

Example: Device Discovery

Dev A			DevB		
Seq No.	Direction	Command /response	Seq No.	Direction	Command /response
1.	Host<-eBMU	**ROK**			
			2.	Host<-eBMU	**ROK**
			3.	Host->eBMU	**AT+JDIS**=3
			4.	Host<-eBMU	**OK**
			5.	Host->eBMU	**AT+JRLS=**1101,11,Serial port,01,000000
			6.	Host<-eBMU	**OK**
7.	Host->eBMU	**AT+JDDS=**0			
8.	Host<-eBMU	**OK**			
9.	Host<-eBMU	**+RDDSRES=**0003199E8B25, devB,000000			
10.	Host<-eBMU	**+RDDSRES=**001842E869D5, Javier N95,5A020C			
10.	Host<-eBMU	**+RDDSCNF=**0			

Example: Service Discovery

Dev A			DevB		
Seq No.	Direction	Command /response	Seq No.	Direction	Command /response
1.	Host<-eBMU	**ROK**			
			2.	Host<-eBMU	**ROK**
			3.	Host->eBMU	**AT+JSEC=1,1,1,04,1111**
			4.	Host<-eBMU	**OK**
			5.	Host->eBMU	**AT+JDIS**=3
			6.	Host<-eBMU	**OK**
			7.	Host->eBMU	**AT+JRLS=**1101,11,Serial port,01,000000
			8.	Host<-eBMU	**OK**
9.	Host->eBMU	**AT+JSEC=1,1,1,04,1111**			
10.	Host<-eBMU	**OK**			
11.	Host->eBMU	**AT+JSDS=**0003199E8B25,1101			
12.	Host<-eBMU	**OK**			
13.	Host<-eBMU	**+RSDSRES=**Serial port,01			
14.	Host<-eBMU	**+RSDSCNF=**0			

Example: Connect with security mode 1 (No PIN request)

DevA Security mode 1			DevB Security mode 1		
1.	Host<-eBMU	**ROK**			
			2.	Host<-eBMU	**ROK**
			3.	Host->eBMU	**AT+JSEC=1,1,1,04,1111**
			4.	Host<-eBMU	**OK**
			5..	Host->eBMU	**AT+JDIS**=3
			6..	Host<-eBMU	**OK**
			7..	Host->eBMU	**AT+JRLS=**1101,11,Serial port,01,000000
			8..	Host<-eBMU	**OK**
			9.	Host->eBMU	**AT+JAAC=**1
			10.	Host<-eBMU	**OK**
11.	Host->eBMU	**AT+JSEC=1,1,1,04,1111**			
12.	Host<-eBMU	**OK**			
13.	Host->eBMU	**AT+JCCR**=0003199E8B25,01			
14.	Host<-eBMU	**OK**			
15.	Host<-eBMU	**+RCCRCNF=**350,0			
			16.	Host<-eBMU	**+RCOI**=0003199E8B35
			17.	Host<-eBMU	**+RCCRCNF=**350,0

Example: Connect with security mode 3 (With PIN request)

DevA Security mode 3 (default)			DevB Security mode 3 (default)		
1.	Host<-eBMU	**ROK**			
			2.	Host<-eBMU	**ROK**
			3.	Host->eBMU	**AT+JDIS**=3
			4.	Host<-eBMU	**OK**
			5.	Host->eBMU	**AT+JRLS=**1101,11,Serial port,01, 000000
			6.	Host<-eBMU	**OK**
			7.	Host->eBMU	**AT+JAAC=**1
			8.	Host<-eBMU	**OK**
9.	Host->eBMU	**AT+JCCR**= 0003199E8B25,01			
10.	Host<-eBMU	**OK**			
11.	Host<-eBMU	**+RPCI=**0003199E8B25			
12.	Host->eBMU	**AT+JPCR=**04,0000			
13.	Host<-eBMU	**OK**			
			14	Host<-eBMU	**+RPCI=**0003199E8B35
			15.	Host->eBMU	**AT+JPCR=**04,0000
			16.	Host<-eBMU	**OK**
			17.	Host<-eBMU	**+RSLE**
			18.	Host<-eBMU	**+RCOI**=0003199E8B35

			19.	Host<-eBMU	**+RCCRCNF=**350,0
20.	Host<-eBMU	**+RSLE**			
21.	Host<-eBMU	**+RCCRCNF=**350,0			

References

[1] www.bluetooth.org

Terminology

A	
APPL	Application
AT	Attention (from Hayes command set)
B	
BT	Bluetooth
C	
CR	Carry Return
CTS	Clear To Send
D	
DEC	DECimal
E	
eBMU	embedded BMU
G	
GPIO	General Purpose Input Output
H	
HEX	HEXadecimal
HW	Hardware
I	
I/O	Input/Output
I2C	Inter-Integrated Circuit
L	
LF	Line Feed
M	
MSB	Most Significant Bit
MTU	Maximum Transmission Unit
O	
OSC	OSCillator
P	
PU	Pull Up
R	
RFCOMM	Radio Frequency Communication
RTS	Request To Send
RX	Reception
S	
SCL	Serial CLock
SDA	Serial DAta
SPP	Serial Port Profile
SW	Software
T	
TX	Transmission
U	
UART	Universal Asynchronous Receiver Transmitter
Z	
Z	Tri-state

This document has been formally released by DOC department (IMC-DOC@intel.com) on September 19, 2011